# Bibliotheca Anonoma Eikonos

Image Semantic Tag Database for Internet Folklife

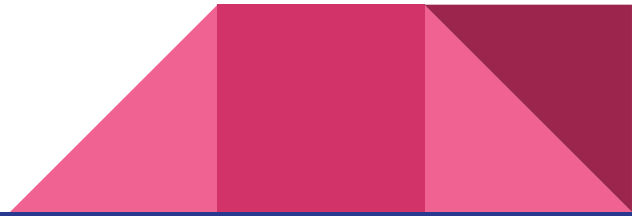
# The Bibliotheca Anonoma

*The Bibliotheca Anonoma is a research library tasked with collecting, documenting, and safeguarding the grand legacy of Internet Folklife: The shared experiences of mankind in a limitless digital network, a virtual universe which has engendered civilizations, culture, trade... and warfare.*

- A research organization I founded for studying and archiving Internet Folklife and History
- Some say that the Internet never forgets: **but it does**
- Like the historical Silk Road, the Internet has an immense effect on human history and global culture
- Like those trade routes, studying Internet history without archived Internet Data is like studying the ocean from a beach

# The Problem

- Archival
  - Many significant images in Internet Folklife from 2003-2008 have been lost to time (Photobucket, Imageshack, Megaupload), hard drive space was expensive
  - Generous donors have given us rare images from these eras that need to be processed.
- Metadata
  - It’s one thing to grab big data. It’s another to find what you want.
  - A tag database for our members to upload their images, and find them afterwards
- Analysis
  - How to find an image when you only know certain categories it matches and elements it has?



# Bibliotheca Anonoma Eikonos - Image Tag DB

- Semantic Image Tag Database
  - Inspired by powerful image tags on Flickr, Pixiv, Danbooru/Gelbooru/Yande.re
  - Multiple Tags narrow down what you want
- Danbooru Engine
  - Used by Konachan, and maintained in production.
  - An image tag database in Ruby with all the bells and whistles. Requires PostgreSQL.
  - Supports multiple categories, pools, tag inheritance
  - Similar images/dupes can be discovered using similar tag discovery

# Proposed Tag Database Structure

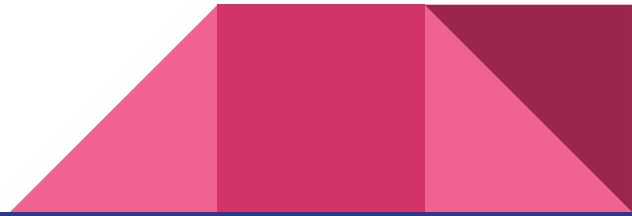
- **Reaction Images** are anything used by 4chan in lieu of words. Because a picture is a thousand words. Reaction images should have tags describing what feeling it describes.
- **Screencaps** will need to be processed with OCR. We could even set up Google Custom Search or our own Sphinx server.
- **Stories** should have a story tag, and possibly a multi-thread post tag. Multiple screencap images should be put in a pool.
- **Greentext** stories, along with whatever replies there are, should be OCR processed and have a dedicated viewer.
- **Memes** are remixable, evolvable, and propogatable ideas or inside jokes that get across the Internet. Tag Inheritance is helpful.

# Step 1: Extract the Data - Macrochan

- First Step: Get some pretagged images
- Macrochan.org - an Image Tag database containing 43,575 images primarily from 4chan's earliest days.
- No longer active, no upload, poor design
- Uses a complex hierarchical tag system that didn't actually implement inheritance.
- Created a scraper to archive entire site with tags

# Step 1: Extract the Data - Dagobah

- Need to extract some pretagged Flash animations
- Flash animations were popular artforms from 1997-2009
- Grave danger of fading away forever
- Dagobah is a pretagged Flash Tag Database
- Created a Python web scraper to sift through poorly designed HTML with pagination
- Needed brute force Positional Scraping



# Step 2: Prepare the Infrastructure

- **YSVPS** (Image Data) - This $15 per month Dreamhost VPS has a grandfathered plan that allows it to store a virtually unlimited amount of data (~2TB). This way, we can store images here and not elsewhere.
  - For the image upload system, it should upload directly to the YSVPS from the client, and not through the main webserver.
- **Amazon RDS pSQL** - PostgreSQL can be run on Amazon AWS instead of Dreamhost for increased performance and safety.
- **Amazon EC2** - If I am using Amazon RDS, I might as well use EC2 to host the Danbooru engine. Though Amazon loves to charge for bandwidth.
  - Paid for using 1 year free trial of Amazon Educate

# Step 2: Upload 700GBs to Internet Archive

- The Bibliotheca Anonoma has S3 Upload access to the Internet Archive
- We made a deal with admins to back up entire two websites: 4plebs.org and RebeccaBlackTech
- Taking up all space on YSVPS, couldn’t figure out how to push data out easily
- Current S3 Upload mechanisms were ineffective:
- Spent weeks attempting to figure it out before I decided to make our own
- Created EZ-IAS3 Upload: Quick and dirty CURL-based Internet Archive S3 Uploader
- Needed a lot of study on HTTP headers and the S3 API
- Pushed out 200GBs of data in one afternoon

# Step 3: High Security on Production Server

- The Internet is famous for random vendettas for whatever reason or another.
- Significantly heightened security for SSH access is required
- Passwords for Authentication: unsafe, well known SSH exploits against password hashes
  - 8 characters is not enough: 16 characters is minimum security these days
- Solutions
  - Use SSH public/private keypair. Store SSH keys on OpenPGP smart card to prevent evil software from scraping keys
  - OpenPGP smart card also only needs PIN upon activation: thus, passwords not needed
  - Use Two-Factor Authentication

# Roadblocks

- First two prerequisites took longer than I thought
- Downloading 43,575 images takes a while
- Positional Scraping is hard work to figure out
- Immense logistical nightmare to push 700GBs of data across the Internet (that’s why I procrastinated)
- By the time I finished downloading data and exporting previously obtained data, it’s the due date

# Plans for the Near Future

- Experiment with Danbooru Engine using Cloud9
  - Investigate alternatives, such as Gelbooru (PHP), Moebooru (improved Ruby fork), PyBooru, etc.
- Distribute Database, Storage, Rails Backends among three servers.
- Implement OCR in the Danbooru Engine (processes text in screencaps)